Recursive Bayesian Filtering Through a Mixture of Gaussian and Discrete Particles

by

Ahmad Saud Manzar

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Master of Applied Science

Queen’s University
Kingston, Ontario, Canada
November 2017

Copyright © Ahmad Saud Manzar, 2017
Abstract

Conventional solutions to nonlinear filtering problems fall into two categories, deterministic and stochastic approaches. While the former is heavily used due to low computational demand, approximation error is tied to their initialization, which causes difficulty during long term application. The latter circumvents this but at the cost of a significant increase in computation. An extremely popular stochastic filter termed the particle filter is especially notorious for this. However its superior performance (over the conventional nonlinear filters) and generality of use makes it ideal in environments where high nonlinearity plagues the state-space model. Estimation error and computational complexity for the particle filter are both related to the number of particles utilized. Yet, many researchers have observed that particles in the vicinity of one another, perhaps because they represent the same state, might be redundant. A new type of filter is proposed where particles in addition to a (linearized) Gaussian component are tracked. This can be seen as a parallel solution to the estimation problem, each component can be separately filtered and constituent outputs summed up to form the filtering distribution. This new filter is then used in two classical scenarios used to benchmark nonlinear filters.
Acknowledgments

In the Name of Allah - the Most Compassionate, Most Merciful.

The completion of this thesis was by no means an entirely individual achievement. I want to began by expressing my gratitude to my supervisor, Dr. Saeed Gazor, for in many ways helping me discover this topic. It was an incredibly difficult journey, both adjusting to the new responsibilities of graduate studies, and the jump in academic ability required. Through all these obstacles my conversations with Dr. Gazor never failed to motivate me to pursue new directions.

Next, I want to thank my professors at Queen’s University, from whom I learned many of the topics that intrigued me and were crucial to my research. Dr. Steven Blostein, for his Estimation & Detection course, which inspired me in my very first semester. Dr. Wai Yip Geoffrey Chan for his Signal Processing & Statistical Learning course, Dr. Il-Min Kim, for his course on Digital Communications, and Dr. Saeed Gazor for his graduate level Probability course. I also had the privilege of auditing several courses, Dr. Glen Takahara’s Discrete Stochastic Processes, Dr. Xiang Li’s Optimization, Dr. Wai Yip Geoffrey Chan’s Multimedia Signal Processing course, and Dr. Aboelmagd Noureldin’s Integrated Navigation Systems, all of which helped reinforce my understanding and expand my creativity.

My friends offered consolidation and advice when I ran into both technical and
other problems. Mohamed Shoeb, I have to thank first, for not only listening to my endless stream of questions but for his friendship and guidance, which made my time in Kingston so enjoyable. Philip Oni, Mohammadali Hedayati, Siavash Eftekharifar, Bicheng Liu, Amr Elwakeel, Mahmoud Ragab, Ashraf Abosekeen, Dr. Hesham Farahat, Hossein Khonsari, Toritseju Okpotse, and Dr. Khaled Al-Hayajneh, thank you all, for lifting my spirits when I was exhausted.

One of my most rewarding experiences was teaching tutorials for Signals & Signals to the class of 2017. This was my favourite course during undergrad and the class honoured my teaching enough to vote me the best third year teaching assistant, an award that I am immensely proud of.

Central to the Queen’s ECE community is our administrative staff. Therefore I want to thank Debbie Fraser, Cheryl Wright, Mary Gillespie, John McKay, Irina Pavich, and Grier Owen for helping make our department run so smoothly.

Last, but certainly not least, I would like to thank my parents and my brother, who will always be my role models. My parents have encouraged me throughout my life, and my brother has taught me more than I could have ever learned myself. They taught me the necessity for and the privilege that knowledge is. It is through knowledge that we become more aware of the world, understand that which confounded us but now makes sense, and ultimately makes us more human. Education is the ideal to which all humans should strive towards, for teaching me that I am forever indebted to them.

I also want to thank NSERC for generously funding me through an Alexander Graham Bell Canada Graduate Scholarship, during my final year.
Contents

Abstract i
Acknowledgments ii
Contents iv
List of Tables vi
List of Figures vii
Chapter 1: Introduction: Recursive Bayesian Estimation 1
  1.1 Bayesian Estimation: ............................................. 1
  1.2 Stochastic Filtering Problem: .................................. 3
  1.3 Review: General Recursive Bayesian Estimation ............... 4
  1.4 Non-linear Recursive Estimation Techniques ................... 4
  1.5 Motivation ......................................................... 7
  1.6 Problem .......................................................... 8
  1.7 Thesis Statement .................................................. 8
  1.8 Contributions ..................................................... 9
  1.9 Organization of Thesis .......................................... 10
Chapter 2: Background 11
  2.1 A Cohesive Overview of Two Approaches to Nonlinear Filtering . 11
  2.2 Previous Attempts at Adaptation in Nonlinear Filtering .......... 14
  2.3 Gaussian Sum Filter .............................................. 15
    2.3.1 Approaches to Gaussian Mixture Reduction ................. 21
  2.4 Generalized Mixture Filtering .................................. 24
  2.5 Solutions based on Sampling Methods ........................... 25
  2.6 Sequential Importance Sampling ................................ 26
  2.7 Pitfalls Associated with Sampling Based Solution Methods ..... 27
  2.8 The Methodology for our Solution ............................ 28
List of Tables

3.1 Expectation Maximization simulation parameters . . . . . . . . . 33

4.1 Simulation parameters for benchmarks . . . . . . . . . . . . . . . 44
List of Figures

4.1 Sample Trajectory with Filter Estimates .......................... 47
4.2 RMSE Over 100 MC runs, $N = 1,000$, $N_{APKF,avg} = 386$ .......... 48
4.3 Percentage of Total Particles used in APKF .......................... 48
4.4 RMSE Over 100 MC runs per timestep, $N = 25,000$, $N_{APKF,avg} = 4,997$ 49
4.5 Sample Trajectory with Filter Estimates .......................... 50
4.6 Trajectory Turned, zoomed, with Filter Estimates .................. 50
Chapter 1

Introduction: Recursive Bayesian Estimation

1.1 Bayesian Estimation:

Central to many fields is the problem of estimating the state of a system, which is a set of variables that completely describe it at any instant of time up to and including the present, given noisy observations [28]. A probabilistic graph model relating the latent and the observed variables is usually used to describe the system. Such a model is known as a state-space representation of the system, and examples of both, additive, and non-additive state-space models are shown below. Although there exists both a continuous and discrete representation of the state space, throughout this thesis only a discrete state space will be considered and referred to. Details concerning such a discretization or validity of model will be overlooked. The parameters $x_k$ denotes the state vector, $w_k$ the process noise, which are both $n \times 1$ dimensional, $z_k$ denotes the measurement vector, and $v_k$ the measurement noise, both $m \times 1$ dimensional, all at discrete time index $k$. The functions $f_k$ and $h_k$ are the process and observation models, respectively.
1.1. BAYESIAN ESTIMATION:

Definition 1. A general nonlinear state-space model is defined as follows,

\[
x_{k+1} = f_k(x_k, w_k) \\
z_k = h_k(x_k, v_k)
\] (1.1)

Definition 2. A special case, is the nonlinear state-space model with additive noise which is defined as follows,

\[
x_{k+1} = f_k(x_k) + w_k \\
z_k = h_k(x_k) + v_k
\] (1.2)

Being the optimal (in the Minimum Mean Squared Error sense) for linear and Gaussian state-space models, Kalman filtering enjoys a wide utilization across many domains. For nonlinear systems, with additive Gaussian noise, the extended Kalman filter has found numerous applications, and is often utilized even in situations where many of the underlying assumptions are violated [28].

Concerning the most general state-space model, the only applicable framework that exists is the particle filtering approach. Though not completely free of issues, such as particle degeneracy in Sequential Importance Sampling, and sample impoverishment in Sequential Importance Resampling, both which will be covered later, are negligible when compared with its vast applicability [2]. The bottleneck for particle filters has always been the computational effort required to solve anything more than toy examples. For computationally and hence energy sensitive applications, particle filtering has seldom even been an option [15]. The work put forth attempts to find a middle ground between the strength of particle filtering, and the modest complexity of a Kalman filter solution.
Proceeding forward, this introduction will define the stochastic filtering problem, and review the recursive Bayesian estimation solution method. It will be at once apparent, that other than for special cases, such as the linear Gaussian case, these equations are intractable. Then attention will be shifted to the general state-space model, and numerical solution methods will be discussed. It is within one of the numerical techniques, that this thesis proposes its innovation.

1.2 Stochastic Filtering Problem:

The stochastic filtering problem can be defined in several different ways, some more general than others. However the form used in this thesis specifically is described. Given a state-space representation of a model, the stochastic filtering problem seeks to compute a single entity, which is called the filtering density $p(x_k|Z_k) = p(x_k|\{z_1, z_2, ..., z_k\})$. (1.3)

From this density, any number of statistical measures can be calculated, subject to some error criterion. Two widely used error criteria are the minimum mean square error (MMSE) and the maximum a posteriori (MAP). Thus subject to either criteria, a point estimate of the state can be calculated, noting that $E(\cdot)$ represents the expectation operator with respect to the density as its argument, via either

$$\hat{x}^{MMSE} = E[x_k|Z_k] = \int x_k p(x_k|Z_k) dx_k$$ (1.4)
\begin{equation}
\hat{x}_{MAP} = \arg \max_{x_k} p(x_k | Z_k)
\end{equation}

1.3 Review: General Recursive Bayesian Estimation

In the hope of keeping this thesis self-contained, the two-steps involved in Bayesian estimation are first reviewed. After initializing the algorithm with a prior density, for the rest of the observation period, said prior is cycled through, first the prediction step and then the update step.

**Algorithm 1 Generalized Recursive Bayesian Estimation**

1: Initialize the algorithm with a prior density on the state,

\begin{equation}
p(x_0 | Z_0) = p(x_0)
\end{equation}

2: **Prediction:** Compute the prediction integral, which gives the A Priori density ,

\begin{equation}
p(x_k | Z_{k-1}) = \int p(x_k | x_{k-1}) p(x_{k-1} | Z_{k-1}) dx_{k-1}
\end{equation}

3: **Update:** Compute the measurement update, which gives the A Posteriori density,

\begin{equation}
p(x_k | Z_k) = \frac{p(z_k | x_k)p(x_k | Z_{k-1})}{\int p(z_k | x_k)p(x_k | Z_{k-1}) dx_k}
\end{equation}

1.4 Non-linear Recursive Estimation Techniques

Faced with the absence of closed form solutions that yield finite dimensional filters, the only approach forward is to utilize numerical approximation techniques. Two
broad classes of approximation techniques exist, those that rely on analysis, and are henceforth known as deterministic, and those that rely on stochastic schemes. Analytic techniques utilize linearization to transform a nonlinear problem around an operating point, to be linear, a common example is the EKF, and a single cycle is presented below.

**Algorithm 2 Extended Kalman Filter [25]**

1. State space representation is given as follows,

   \[ x_k = f_{k-1}(\hat{x}_{k-1}, w_{k-1}) \sim \mathcal{N}(0, Q_k) \] (1.9)
   
   \[ z_k = h_k(x_k, v_k) \sim \mathcal{N}(0, R_k) \] (1.10)

2. **Initialization:** With prior state density,

   \[ p(x_0) = \mathcal{N}(\hat{x}_0^+, P_0^+) \]

   (1.11)

3. **Time Propagation:**

   \[ \hat{x}_k^- = f_{k-1}(\hat{x}_{k-1}^+) \] (1.12)
   
   \[ P_k^- = F_{k-1} \hat{P}_{k-1}^+ F_{k-1}^T + W_{k-1} + Q_{k-1} W_{k-1}^T \] (1.13)

4. **Measurement Update:**

   \[ H_k = \frac{\partial h_k}{\partial \hat{x}}(f_{k-1}(\hat{x}_{k-1}^+), 0) \] (1.14)
   
   \[ V_k = \frac{\partial h_k}{\partial v}(f_{k-1}(\hat{x}_{k-1}^+), 0) \] (1.15)
   
   \[ K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \] (1.16)
   
   \[ \hat{x}_k^+ = \hat{x}_k^- + K_k [y_k - h_k(\hat{x}_k^-)] \] (1.17)
   
   \[ P_k^+ = (I - K_k H_k) P_k^- \] (1.18)
1.4. NON-LINEAR RECURSIVE ESTIMATION TECHNIQUES

Stochastic techniques utilize sampled points which are known as particles, originating from a density known as the proposal, then weighting these particles to approximate the filtering density, a common example is the particle filter, a single cycle is presented below.

Algorithm 3 Bootstrap Particle Filter [28]

1: State space representation is given as follows,

\[ x_{k+1} = f_k(x_k, w_k) \]  \hspace{1cm} (1.19)
\[ z_k = h_k(x_k, v_k) \]  \hspace{1cm} (1.20)

2: **Initialization:** With prior state density, \( p(x_0) \), and draw \( N \) samples, denoted by \( \{x^+_{0,i}\}_{i=1}^N \) which are known as particles.

3: **Time Propagation:**

\[ x^-_{k,i} = f_{k-1}(x^+_{k-1,i}, w_{k-1}^i) \]  \hspace{1cm} (1.21)

4: **Measurement Update:** compute the relative likelihoods also known as weights,

\[ q_{k,i} = p(z_k|x^-_{k,i}) \], then normalize them. In some contexts this is known as the estimation step.

5: **Resampling:** Draw a set of a posteriori particles on the basis of these weights, that is to say the transition from a priori to a posteriori occurs with a probability of \( q_{k,i} \). This set constitutes a discrete approximation to \( p(x_k|Z_k) \).

The main focus of this thesis will be stochastic approximation schemes which are known more commonly as Monte Carlo techniques. Within this broad class of sampling methods, the subset of which can be implemented sequentially are utilized [5]. While this may seem like a small subset, sequential Monte Carlo methods (SMC)
have resulted in remarkably accurate results when applied to nonlinear filtering [10].
In Chapter 2, details of this family of techniques will be given along with some variants
used specifically in this thesis.

1.5 Motivation

A major concern when using sampling based solution strategies, whether sequen-
tially implemented or otherwise, is the large amount of computational resources re-
quired. Since a discrete approximation by default requires a larger number of points
to accurately describe a distribution, and with the presence of artefacts such as sam-
ple impoverishment, frequently a very large number of particles is specified [28]. Yet
there has been significant belief that particles (which form the discrete delta approx-
imation) within the vicinity of one another in the state-space should in fact represent
the same state, and are hence redundant [15].

The idea of a local approximation is not unique or even revolutionary. Local ap-
proximation methods such as the Gaussian sum filter have been around for decades.
However what rules out their application (in general state-space models) are issues
related to unbounded complexity and their tendency to diverge globally [1]. In se-
quential Monte Carlo techniques, such local approximations serve different purposes
(used to approximate the optimal sampling density), however they still positively
influence the estimator tractability. It is along this line of thought that motivated a
method where the amount of particles could be kept at a minimum while keeping the
estimator quality fixed (to that of a purely particle approximation).
1.6 Problem

Approximate solutions for generalized Bayesian recursion are the only implementable solutions for practical problems. However in many cases they require an astounding amount of computation. Hence a computationally efficient solution has long been sought, one that was not problem specific (such as the Rao-Blackwellized particle filter, which requires a stringent factorization of the state-space model) [27]. In addition, there is hope that by monitoring the nonlinearity, such an estimator would be able to adapt, using a combination of the extended Kalman filter for Gaussian components, and the particle filter for highly nonlinear regions. This thesis presents one such solution.

1.7 Thesis Statement

The aim of this thesis is to show that using intuition related to the spatial distribution of particles, an efficient filtering algorithm can be defined, one which preserves the low computational cost of Gaussian filters, with the tracking ability of particle filters. Through this, it is hypothesized that even with a modest number of particles, estimation quality can be maintained while computation is reduced.

Assessing state estimator quality is a difficult task, depending on the underlying system certain algorithms perform better than others. The main measure of estimator performance is the root mean squared error, while many have questioned its accuracy in multimodal problems, its use is sufficiently standard in the nonlinear filtering community to, in the least be used as a qualitative indicator.
1.8 Contributions

In this thesis, we have shown that there exists a technique to reduce the number of particles utilized in a standard bootstrap filter, through the introduction of a Gaussian component which is tracked throughout using a nonlinear (Gaussian) filter. Based on the belief that a portion of the state density displays Gaussian characteristics, the remainder can then be approximated via Monte Carlo sampling i.e. particles. This ensures that an excess of particles, that might otherwise represent a Gaussian component are not wasted, and instead assigned a suitable mean vector and covariance matrix.

This is achieved through the development of a new filter, and as this filter is the focus, a brief description of one cycle is summarized here. After samples are drawn from the prior state distribution, resolution into a mixture of Gaussian density is achieved using the expectation-maximization algorithm, a standard tool to iteratively estimate required parameters. The mixing coefficient which has the larger weight, is taken as the Gaussian, and all data points that are not most likely part of this component are assigned as particles. Then a posterior probability is calculated for each point conditioned on the two classes. Proceeding forth, the particles are filtered using particle methodology, and the Gaussian is filtered using any nonlinear filter. Finally the estimate for the posterior distribution (states given observations) is made through a linear combination of the estimates for each filter weighted by the their mixture coefficients.
1.9 Organization of Thesis

Moving forward, background pertinent to the all the major sub areas covered in the thesis is presented concisely and in a structured manner in Chapter 2. It is important to connect the motivation with solutions mentioned in the literature, categorizing them into a solution framework. This framework has two main branches, one that deals with mixture filtering approaches and one that deals with importance sampling approaches. Once this framework is understood our approach will be presented in detail, highlighting shortcomings in previous work that our algorithm improved upon in Chapter 3. Results and evaluation will be presented in the following chapter, benchmarking our algorithm on classically hard examples. To conclude the thesis, future work and conclusions are presented highlighting its accomplishments and viable future paths of inquiry.
Chapter 2

Background

2.1 A Cohesive Overview of Two Approaches to Nonlinear Filtering

Real-time nonlinear filtering has become essential in this age, with the proliferation of smartphone sensors. Since tasks that were once reserved to defense or large-scale operators are now required by the everyday user, e.g. indoor localization and GPS/INS based navigation, an efficient and power sensitive set of algorithms needs to be devised [17]. Sensors are often modelled nonlinearly, with respect to the quantity they are measuring and along with drift and other accumulating error mechanisms, straightforward linearization is not the way forward [20]. Instead filters that are at their heart nonlinear, such as the particle filter are necessary for improved tracking [31]. However before such a goal can be realized, it is wise to survey previous attempts within the community to deal with similar issues.

The Gaussian sum filter (GSF) is one such attempt. Devised by Alspach et al. in an incredible and simply motivated paper [1], the authors sought to generalize the extended Kalman filter to systems where it was known to diverge. Their method
2.1. A COHESIVE OVERVIEW OF TWO APPROACHES TO NONLINEAR FILTERING

was as follows; if the initial set of densities can be represented as a mixture of Gaussian densities, then applying an EKF to each component would work better than tracking a single Gaussian. The result is an algorithm that is essentially a filter bank, where individual filters are all EKFs, linearized about each mixture component mean. Therefore, instead of a global linearization, it takes place around smaller neighbourhoods and in theory is much better. GSF brought a lot of freedom, no longer did the assumptions on noise and initial density being Gaussian needed to be enforced, instead with a mixture representation of either, the classical EKF equations were all still valid, but with minor modifications.

Central to any filtering technique is propagation and update steps, in the framework of the GSF, these are termed mixture propagation, and mixture update. One additional aspect needs to be added, which is the mixture maintenance. In their original paper, Alspach et al. mentioned the tendency of the GSF to keep components with extremely low weights and continually add on components as the estimation proceeded forward. The first point is remedied through a set of techniques known as Gaussian mixture reduction [1]. The second point is isolated in the cases where the process and observation noise are not adequately represented by a single Gaussian component [29]. Both of these will be discussed later, since understanding them is crucial to mixture filtering as a whole. With the GSF a certain aspect of adaptability was introduced, while still using an analytically linearized filter in the background. Since then, the GSF has been generalized to use several other local filters, such as the unscented Kalman, and the Gauss-Hermite filters, but neither of these are purely nonlinear. [14].

Perhaps the only purely known nonlinear filter is the particle filter (PF). In its
2.1. A COHESIVE OVERVIEW OF TWO APPROACHES TO NONLINEAR FILTERING

essence the PF is an intelligent version of a very well-known computationally dense filter, the point-mass or grid filter. PFs use a probabilistic approach to grid selection and are able to guide grid points into regions of nonlinearity much better than by using a uniform grid. Yet, the bottleneck for PFs has always been the large number of starting particles needed to prevent sample impoverishment [23]. Improvements have been proposed such as the regularized PF but additional complexity is introduced by means of a kernel density estimation in the resampling step, which in many cases does not improve the estimator quality, and increases computation [2].

Importance sampling, the statistical tool which allows the evaluation of an otherwise intractable integral via Monte Carlo integration is almost entirely dependent on the sampling density used. Indeed optimal sampling densities can be derived, though since intractable for the majority of cases and are even more useless. Two significant cases where a closed form optimal sampling density can be calculated are in the linear and additive Gaussian setup, and the nonlinear state with linear observation model setup. For the first case, Kalman Filtering already exists, and for the other, there is not much practical value since sensors are usually nonlinear in their observation equation. Thus seeking a closed form optimal sampling density is out of the question. Approximate sampling densities can be calculated by approximation of the optimal sampling density via a nonlinear filter, such as the unscented Kalman filter (UKF) or Gauss Hermite Kalman filter (GHKF), and results in algorithms that are known in the literature as unscented particle filter, and Gauss Hermite particle filters, respectively. One major downside to such sampling densities is that they can make the filter unstable, and hence should be used with caution [25].
2.2 Previous Attempts at Adaptation in Nonlinear Filtering

Being a well established tool, avenues for research might seem few in the PF literature, but note the lack of work within the area of a sub problem, namely adapting the particle set size. To the author’s knowledge, the only paper that has worked on this problem was recently published [9], which itself utilizes theory developed half a decade earlier by the same authors [8]. On-line adaptation (for the number of particles) is by no means an easy problem, since one needs to assess the quality of a dynamical system model during the course of estimation. However taking a step back, Liu et al proposed an EKF and UKF switching filter, where the accuracy of estimation was fulfilled using a switching mechanism related to the trace of the previous iteration covariance matrix [19]. This was needed because the EKF is known to diverge quickly in some cases, yet its computations are fast. Thus if there does not exist a need for using something more robust such as the UKF, the former is a perfectly fine choice. Liu et al used the UKF because they felt the particle filter was not usable for their application in real-time.

Returning to the particle filter, a significant amount of work was done on the resampling step, which is one of the most costly. Deterministic resampling was a method proposed by Li et al [16] where the spatial distribution of the particle set is fully utilized, instead of just the weights, as is the case with traditional resampling methods. While this might seem like an improvement, Li et al also noted that such a technique eliminates the dimensionally invariant property, said differently such a resampling strategy is easier in lower but not so in higher dimensions. A second issue which arose is the need for particle splitting, since the initial part of deterministic resampling clustered particles, before the following time propagation step, particles
needed to be plentiful enough to explore the state space. Again this adds computation not traditionally required. Thus with these concerns, it is not surprising that deterministic resampling is not widely used. Where it is used, such as in mobile robot localization, it works very efficiently [15].

It is important to remember that resampling is not the same as the estimation step in particle filters. Resampling is the technique used to curb the particle degeneracy problem that plagued sequential importance sampling (SIS) [10]. Enough emphasis cannot be placed on how essential this step is. To put it bluntly, when there are particles (in the context of nonlinear filters) one must perform resampling. Thus the next logical progression is to focus on the estimation step for possible innovations.

2.3 Gaussian Sum Filter

One of the earliest variants of the Kalman filter applied to nonlinear problems, was operating under the assumption that the posterior distribution could be represented parametrically as a mixture of Gaussian density. Thus once the state-space model was linearized, the extended Kalman filter equations can be used to update the mean and covariance for individual components. The a posteriori state estimate is then formed through taking a weighted average of the individual means, weighted through the mixing coefficient [28].

One specific problem observed with this approach was when the process and the observation noise were represented with more than a single Gaussian. In such a case, an exponentially growing number of terms needed to be updated. These terms are explicitly tied to Gaussian mixture representations of noise, when a single Gaussian component is sufficient, no growth in number of components is observed [29], such a
version is presented in algorithm 4.

A similar issue arises during deployment, certain components with extremely small component weights are updated even though they factor little into the final estimate. Mixture reduction techniques is the sub area that concerns itself with maintaining the shape of a density while reducing the number of components used [6].

Algorithmically, the GSF is a conceptually easy variant to understand. It can be viewed as a parallel bank of non-linear filters working in tandem. Even though [1] used the extended Kalman filter, any type of filter based on a linearization scheme, be it analytic, as is the case of the EKF or statistical, as is the case of the UKF, either could be used. Of course decomposition of an arbitrary density into a mixture of gaussian density is not a trivial task, but can be accomplished via the expectation maximization algorithm, among others. Such a decomposition usually has to be carried out for three quantities, the initial state, process noise, and observation noise densities [29].

Note that in the discussion that follows, and throughout this thesis, a priori estimates will be denoted with a minus sign, while a posteriori estimates with a plus sign in the superscript. In the subscript, a pair denoting time index and component/or particle respectively is located. For example $\hat{x}^{+}_{k,i}$ is the a posteriori mean at time index, $k$, for the $i^{th}$ component. It should also be stressed that before a measurement is received an a priori estimate cannot transition to an a posterior estimate, at the same time index. However the converse in a sense, that is to say an a posteriori estimate may transition to an a priori estimate, of the following time index, is possible but only through time propagation. Mixing weights, which in this section refer to mixture coefficients in a Gaussian mixture, are left out from the main development since they
have long expressions, and no illustrative value. Mixing weights in the next section refer to particle weights, and their expressions are presented.

Fundamentally the motivation is as follows, linearization techniques allow the use of closed form expressions for state estimation, under the premise that the prior and noise densities can be expressed as a finite sum of Gaussian components, notionally a weight is added to each pdf. The notation $\alpha \cdot \mathcal{N}(\mu, \Sigma)$ denotes a normal distribution with mean, $\mu$ and covariance matrix, $\Sigma$ and $\alpha$, the mixing coefficient. This in turn means that the predictive and filtering densities themselves can be written as a mixture,

$$p(x_k | Z_{k-1}) = \sum_{i=1}^{N_1} \alpha_{k,i}^- \mathcal{N}(\hat{x}_{k,i}^-, P_{k,i}^-)$$

$$p(x_k | Z_k) = \sum_{i=1}^{N_2} \alpha_{k,i}^+ \mathcal{N}(\hat{x}_{k,i}^+, P_{k,i}^+)$$

This then implies that the complexity of the nonlinear estimation problem can be bounded, unlike Monte-Carlo methods, that while general, require expensive calculations. While the ability to express an arbitrary density as a Gaussian mixture model (GMM), is essentially a parameter estimation problem, and this is one route to innovate upon, the GSF has a hidden issue that makes its use impractical, and will be shown in what follows.

Suppose a state estimation problem, at time $k - 1$ has a posterior density that requires $K$ components,

$$p(x_{k-1} | Z_{k-1}) = \sum_{i=1}^{K} \alpha_{k-1,i}^+ \mathcal{N}(\hat{x}_{k-1,i}^+, P_{k-1,i}^+)$$
and the process noise, and observation noise require, $L$ and $M$ components, respectively,

$$p(w_k) = \sum_{j=1}^{L} \beta_{k,j} N(\hat{w}_{k,j}, B_{k,j}). \quad (2.4)$$

Here we can also interject and write an expression for the predicted state density as follows:

$$p(x_k|\mathcal{Z}_{k-1}) = \sum_{i=1}^{K} \sum_{j=1}^{L} \alpha_{k-1,(i,j)}^+ \cdot \beta_{k,j} \cdot N(\hat{x}_{k,(i,j)}^-, P_{k,(i,j)}^-). \quad (2.5)$$

Recall traditional Bayesian prediction is

$$p(x_k|\mathcal{Z}_{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|\mathcal{Z}_{k-1})dx_{k-1} \quad (2.6)$$

It should be noted that the application of a nonlinear filter, such as EKF or UKF for each pair $(i,j)$ of mixture components is required. The mean and the variance are the output of the chosen filter. If an EKF is used for example,

$$F_{k-1} = \frac{\partial f_{k-1}}{\partial x} \bigg|_{\hat{x}_{k-1}^+} \quad (2.7)$$

$$W_{k-1} = \frac{\partial f_{k-1}}{\partial w} \bigg|_{\hat{x}_{k-1}^+} \quad (2.8)$$

$$\hat{x}_{k}^- = f_{k-1}(\hat{x}_{k-1}^+) \quad (2.9)$$

$$P_{k}^- = F_{k-1} P_{k-1}^+ F_{k-1}^T + W_{k-1} Q_{k-1} W_{k-1}^T. \quad (2.10)$$

Proceeding to the update step, begin by representing the observation noise as a
2.3. GAUSSIAN SUM FILTER

Gaussian mixture,

\[ p(v_k) = \sum_{l=1}^{M} \gamma_{k,l} \mathcal{N}(\hat{v}_k, \Gamma_{k,l}). \] (2.11)

Now again recall that in the Bayesian update step, the following terms are calculated,

\[ p(x_k|Z_k) = \frac{p(z_k|x_k)p(x_k|Z_{k-1})}{\int p(z_k|x_k)p(x_k|Z_{k-1})dx_k}. \] (2.12)

Next, choosing a non-linear filter, for each 3-tuple, \((i,j,l)\) of mixture components perform an update step, and the resulting posterior density is as follows,

\[ p(x_k|Z_k) = \sum_{i=1}^{K} \sum_{j=1}^{L} \sum_{l=1}^{M} \alpha^+_{k,(i,j,l)} \cdot \mathcal{N}(\hat{x}^+_k, P^+_k, (i,j,l)) \] (2.13)

where if the EKF is chosen,

\[ H_k = \frac{\partial h_k}{\partial x} \bigg|_{f_{k-1}^-(\hat{x}^-_{k-1})} \] (2.14)
\[ V_k = \frac{\partial h_k}{\partial v} \bigg|_{f_{k-1}^-(\hat{x}^-_{k-1})} \] (2.15)
\[ r_k = y_k - h_k(\hat{x}^-_k, 0) \] (2.16)
\[ S_k = H_k P^-_k H^T_k + V_k R_k V^T_k \] (2.17)
\[ K_k = P^-_k H^T_k S_k^{-1} \] (2.18)
\[ \hat{x}^+_k = \hat{x}^-_k + K_k r_k \] (2.19)
\[ P^+_k = (I - K_k H_k) P^-_k. \] (2.20)

The number of components required to sum over is \(KLM\). However using the same reasoning, the next step will require an additional \(LM\) components, and the final
2.3. **GAUSSIAN SUM FILTER**

step will then require summing over $K L^k M^k$ after $k$ steps of the algorithm. Stated differently, filtering using the GSF requires an exponential number of components. Mixture reduction algorithms were developed following this observation, and were discussed in [29]. Additionally, when dealing with merging components, many of the results require that a pairwise measure is calculated, this means that if $N$ is the total number of components, then $\binom{N}{2}$ quantities have to be analyzed. It also needs to be mentioned that to understand the complexity of reducing an $N$ component mixture, to that of $L$ components, the possible ways of grouping them are described by a Stirling Number of the second kind [6].

When considering the practicality of the Gaussian Sum Filter, specifically in the case of non-Gaussian process (state) and measurement noise, after the measurement update step, more components are present at the end of the cycle than at the start. This poses a significant problem when implementing the GSF. Hence an algorithm must be adopted to reduce mixture components, also given many components finding a single component Gaussian to represent them is also needed, this is often accomplished through the moment preserving merge, which is outlined below.

**Theorem 3. Moment Preserving Merging:** To convert an $L$ component Gaussian mixture into a single component via the following relationships,

$$p(x) = \sum_{i=1}^{L} a_i \cdot \mathcal{N}(\hat{x}_i, P_i) \rightarrow a \cdot \mathcal{N}(\hat{x}, P), \quad (2.21)$$
2.3. GAUSSIAN SUM FILTER

\[ a = \sum_{i=1}^{L} a_i \]  \hspace{1cm} (2.22)

\[ \hat{x} = \frac{1}{a} \sum_{i=1}^{L} a_i \cdot \hat{x}_i \]  \hspace{1cm} (2.23)

\[ P = \frac{1}{a} \sum_{i=1}^{L} a_i \{ P_i + (\hat{x}_i - \hat{x})(\hat{x}_i - \hat{x})^T \} . \]  \hspace{1cm} (2.24)

In general, Gaussian mixture reduction techniques fall broadly into three different approaches, threshold, distance, and global optimization based. The moment preserving merge can be seen as the second part to a routine that first uses a distance-based technique to find components that are near enough to one another, and then combines them. The three major approaches are summarized below [6].

2.3.1 Approaches to Gaussian Mixture Reduction

1. **Threshold-based** also known as pruning. Easily the most widely used in online trackers, given a predefined threshold, \( \epsilon \), all components whose weights are less than this are removed, and the remaining weights are rescaled likewise. This could also be modified to remove all components with a cumulative weight less than \( \epsilon \) as well.

2. **Distance-based**: A metric of some sort is used to calculate distances between components, then components too near to one another are merged. Different metrics have been proposed, the standard is to use the Mahalonobis distance, but benefits have been observed when using the Bhattacharyya or squared-Hellinger distance. This approach is the most widely accepted in terms of
accuracy of final representation versus complexity of decision trade off. One problem that arises with these methods is that lower weight components are merged which might not be an effective choice.

3. **Optimization-based:** The goal is simple, maximize the similarity between the original mixture, and a reduced version of it. Equivalently this amounts to minimizing the number of components, \( N \), the weights, \( \alpha \), and the component means and covariance matrices, \( \mu \) and \( \Sigma \) respectively. While this is usually a daunting task, Williams and Maybeck originally pioneered this idea \cite{6}, and in some sense Runnalls’ algorithm, which uses an upperbound to the Kullback–Leibler divergence, instead of integral square error, was developed in the same line of thought, albeit much more efficient \cite{24}.

It should also be stressed the phenomenon that increases the number of terms throughout the estimation is directly tied to whether process and observation noise can be effectively modelled by a single Gaussian component. If this is the case, then the number of components stays constant, and hence no reduction procedures are required.

The choice of one local filter over another, the filtering algorithm used for each individual component, needs to be understood on a problem by problem basis. That is to say, that with the same number of components, utilizing a EKF does not guarantee superior estimator performance over the unscented Kalman filter or the Gauss Hermite Kalman filter, or vice versa. When choosing a local filter other features such as linearization stability or difficulty in derivative-free based implementation should be considered first, along with storage constraints, and allowable computational delay \cite{29}.
2.3. GAUSSIAN SUM FILTER

Once filtering is completed, the same equations used in the moment preserving merge are also the ones used to compute the MMSE a posteriori state estimate. Thus the MMSE state estimator is calculated using the following relation,

$$\hat{x}^{MMSE} = \sum_{i=1}^{M} a_{k,i} \cdot \hat{x}^{+_i}_{k}$$

(2.25)

where the estimator is a weighted average of the individual component means.

**Algorithm 4** Gaussian Sum Filter Algorithm (constant complexity version) [28]

1: Consider the General State-Space, with

$$x_k = f_{k-1}(x_{k-1}, w_{k-1})$$

(2.26)

$$z_k = h_k(x_k, v_k)$$

(2.27)

$$w_k \sim \mathcal{N}(0, Q_k)$$

(2.28)

$$v_k \sim \mathcal{N}(0, R_k)$$

(2.29)

2: Approximate the the pdf of the initial state, \(p(x_0)\) as a mixture of Gaussians

$$p(x_0) = \sum_{i=1}^{M} a_{0,i} \mathcal{N}(\hat{x}^{+_0}_{i}, P^{+_0}_{i})$$

(2.30)

3: Construct the A priori state-estimate, noting \(a_{k,i} = a_{k-1,i}\), and using EKF time propagation for mean and covariance matrices.

$$p(\hat{x}^{-}_k) = \sum_{i=1}^{M} a_{k,i} \mathcal{N}(\hat{x}^{-}_k, P^{-}_{k,i})$$

(2.31)
2.4. GENERALIZED MIXTURE FILTERING

4: Mixture weights are calculated using the following set of equations,

\[ r_{k,i} = y_k - h_k(\hat{x}_{k,i}^-, 0) \]  \hfill (2.32)
\[ S_{k,i} = H_{k,i} P_{k,i}^{-1} H_{k,i}^T + R_k \]  \hfill (2.33)
\[ \beta_{k,i} = \exp\left(-\frac{r_{k,i}^T S_{k,i}^{-1} r_{k,i}}{2}\right) \frac{1}{(2\pi)^{n/2}|S_{k,i}|^{1/2}} \]  \hfill (2.34)
\[ a_{k,i} = \frac{a_{k-1,i} \beta_{k,i}}{\sum_{j=1}^M a_{k-1,j} \beta_{k,j}} \]  \hfill (2.35)

5: Construct the A posteriori state estimate now, using EKF measurement update for mean and covariance matrices.

\[ p(\hat{x}_k^+) = \sum_{i=1}^M a_{k,i} \mathcal{N}(\hat{x}_{k,i}^+, P_{k,i}^+) \]  \hfill (2.36)

2.4 Generalized Mixture Filtering

In [30] the Gaussian mixture assumption was lifted and formulae were derived considering a general mixture density. They showed that considering a mixture representation of the posterior allowed multiple modes to be tracked effectively, a requirement specifically useful in visual tracking applications. During the derivation of a generalized mixture filtering algorithm, Vermaak et al. [30] showed that for most densities the mixture Bayesian recursion equations were themselves intractable, thus Sequential Importance Resampling (SIR) approximations were required for practical implementation. Obtaining a mixture representation, propagation and maintenance are covered now.
Perhaps the paper’s largest contribution is showing that the mixture of particle filters, (each component within the mixture is updated using SMC) interact only via the mixing coefficients. This paves the way for adaptability when it comes to resampling a given mixture, a single resampling scheme need not be used for all components [30]. Maintenance is also easy, and especially important as several components might be too disperse, causing difficulty in accurate representation of the filtering density or share too great an overlap, meaning wasted resources and hence the mixture will need to be recomputed. The obvious constraint being the new representation cannot represent a different distribution, this is in part insured by the number of particles remaining unchanged during maintenance.

2.5 Solutions based on Sampling Methods

As approximate solutions methods are the only ones that seem to exist for general nonlinear filtering problems, numerical approximations are of paramount importance. To solve the nonlinear filtering problem the following form of integrals is frequently encountered:

\[ E[g(x)|Z_k] = \int g(x) \cdot p(x|Z_k) dx. \] (2.37)

Sampling methods concern themselves with finding points, which are called particles, such that integrals like those above can be reduced to the following form,

\[ \int g(x) \cdot p(x|Z_k) dx \approx \frac{1}{N} \sum_{i=1}^{N} g(x_{k,i}^+). \] (2.38)

The rationale for one such method can be found in Monte Carlo integration, in which
these points are (ideally) drawn independently from the target density. Under this ideal scenario, regardless of the dimensionality of the state vector, $x$, such methods are guaranteed to converge according to the Central Limit Theorem, on the order of $O(\sqrt{N})$ [25].

In practice, sampling from the target density is incredibly difficult, recall conditioning requires an iterated integral thus, than additional integral is added when each observation becomes available. In statistics, a method to circumvent this, importance sampling (IS) is well known. IS samples from a density other than the target and tries to form a Monte Carlo approximation to the target using these samples. Naturally, the quality of this alternative density will translate directly into a better estimate. Referred to as the importance distribution, or the proposal depending in the context, the performance of IS rests to a greater extend on its choice, than the number of samples used [2].

2.6 Sequential Importance Sampling

When utilizing nonlinear filtering in a practical situation, measurements arrive sequentially and an estimate of the state is required as quickly as possible. In such an environment batch processing is not helpful. Sequential importance sampling aims to sequentially implement importance sampling, and to achieve this drawing samples and calculating their weights needs to formulated likewise. Once this is carried out a discrete approximation to the target density is formed as,

$$p(x_k|Z_k) \approx \sum_{i=1}^{N} q_{k,i} \cdot \delta(x_k - x_{k,i}^+).$$

(2.39)
However, while the particles, $x_{k,i}^+$ can be drawn from an importance distribution, a sequential implementation also requires the posterior lend itself to factoring in a specific way. Such a way would make it so that upon the arrival of an observation, the previous state trajectory could be appended to the current state estimate, i.e. recursively. In addition, if a recursive form of the proposal, $\pi(\cdot)$ can be found, then it can be shown that the weights, $\{q_{k,i}\}_{i=1}^N$ obey the following formula,

$$q_{k,i} \propto \frac{p(z_k|x_{k,i}^-)p(x_{k,i}^-|x_{k-1,i}^+)}{\pi(x_{k,i}^-|x_{k-1,i}^+)} \cdot \frac{p(x_{k-1,i}^+|Z_{k-1})}{\pi(x_{k-1,i}^-|Z_{k-1})},$$

(2.40)

where the second term is actually the weights calculated at the previous time [12]. Thus a recursion for the weights can be shown to be as follows,

$$q_{k,i} \propto \frac{p(z_k|x_{k,i}^+)}{\pi(x_{k,i}^+|X_{k-1,i}^+)} \cdot q_{k-1,i}. \quad (2.41)$$

### 2.7 Pitfalls Associated with Sampling Based Solution Methods

The methodology presented thus far was known for many decades before the seminal paper of Gordon[10]. Indeed if SIS was applied to the nonlinear filtering problem, the revitalization of this technique would not have taken place. SIS is known to possess a serious degeneracy problem, a phenomenon where the majority of particle weights tend either extremely close to, or are equivalently zero. The key observation in [10] was the addition of a resampling step, where particles with small weights are removed and those with large weights are replicated. This amendment of the SIS algorithm is so important that it was renamed to reflect this, and called Sequential Importance Resampling (SIR). When the term particle filter is mentioned, this is equivalent to saying the SIR algorithm was used.
2.8. THE METHODOLOGY FOR OUR SOLUTION

Depending on when resampling is administered, for example whether this happens in each step or after some control on the number of particles, such as the effective number of particles drops below a certain threshold, differentiates one variant of SIR from another. An example of the former would be the bootstrap filter, and of the latter, adaptive resampling PF. There are also resampling methods which sample from a discrete version of $p(x_k|Z_k)$, and others which sample from continuous interpolation [15]. A caveat of resampling from a discrete distribution is that it introduces a side effect, which is known as sample impoverishment. This refers to a lack of diversity among particles, since resampling favours those particles with higher weights, in some extreme cases, such as when the process noise is very small, all particles collapse to a single particle. Methods to combat this effect are regularization, roughening or by using an auxiliary particle filter. Again, none of these methods are without additional cost. Regularization for example constructs a kernel density estimate of the discrete distribution before it resamples and this is computationally expensive. Ignoring computational complexity, regularization is also costly when applied to multi-modal densities, since (global) regularization tends to suffer when tracking multiple modes, and encourages uni-modality [21].

2.8 The Methodology for our Solution

Now that a general overview has been provided it is time to look at the development followed in this thesis. The goal of this thesis was to develop a work around, to the high computational complexity that a standard particle filter requires. Through understanding the work that has been carried out in the mixture filtering and Monte Carlo filtering sub disciplines, a filter is proposed as a comprise between the two areas,
taking a little from each camp. Saying that a filter combines the Kalman filter and the particle filter for example, is a nonsensical statement, because the combination methodology has not been specified. There does however exist a very powerful and computationally cheap solution, the Rao-Blackwellized particle filter, which can be thought of a combined Kalman particle filter.

The Rao-Blackwellized particle filter owing its name to a theorem for minimum variance estimators, approaches the problem of filtering by searching for what is known as a conditional Gaussian linear substructure to the state-space model. Such a structure enables the using the Kalman filter for linear states, and using the particle filter for nonlinear states. While the RBPF works extremely well in practice, practitioners are required to find a factorization for the state-space which in many cases is hard to calculate [13]. For cases where such a model can be used all is well, but for those where one does not exist naturally nothing more is achieved. Schon et al [27] published a paper where very general state-space models were decomposed into the conditional Gaussian linear substructure. RBPF is still extensively used in navigation and positioning applications, where model can easily have a state-space of a dozen or more dimensions, for such models the estimator is sped up by quite a bit [12]. Therefore, the take home point is, RBPF is excellent for specific problems but not for the general problem.
Chapter 3

An Adaptive Particle Kalman Filter

3.1 Introduction

Combining the particle and Kalman filter has already been shown to be a very broad statement, one that necessitates further explanation. One such method, the RBPF was discussed briefly with its significant benefits and narrow applicability. In the coming sections, the guiding light for the discussion will be how Gaussian mixture filtering can be combined with importance sampling based methods. First, it is important to discuss the generalization of mixture filtering to non-Gaussian mixture densities. Vermaak et al. elected to approach the problem of multiple mode tracking, an issue of great importance in computer vision. The group wanted to understand which of the assumptions of Gaussian mixture filtering could be weakened, to any valid density. Remarkably they were able to develop the generalization in one conference paper, an article whose contribution was significant but went unnoticed in the general community.
Vermaak et al began by deriving all the equations analytically necessary for estimation, the prediction-update chain to be specific, and eliminated parametric assumptions on the density. What followed was the not so surprising revelation that the equations were intractable, and hence required an application of IS to be computed. Specifically IS was needed to compute the filtering recursion, and to compute component likelihoods. An interesting result they showcased was that the filtering recursion for each component could be carried out separately. Furthermore, when it came time for mixture maintenance, all that was required was changing limits on summations, to get a new representation after a clustering function had been applied. This work more importantly outlined what would be needed in the development of a mixture filtering algorithm [30].

With mixture filtering generalized to any arbitrary continuous density, the question becomes what variants in the PF literature exist that try to generalize its methodology? In many respects the choice of a Dirac delta as a kernel for approximation is the most general it can be. Thus what is presented now are methods that seek solutions to issues in particle filtering. These issues are almost entirely under the umbrella of solutions to sample impoverishment. Sample impoverishment is the side effect of resampling. During resampling, particles can be thought of as transitioning from a priori to a posteriori, and the probability of this taking place is given by (at least in the bootstrap filter) the likelihood. Noting that resampling takes place after the measurement update, it is carried out from a discrete distribution, and here is what causes impoverishment. One effective solution is to sample from a continuous distribution, and this motivates the regularized particle filter.

Regularized particle filter (RPF) is exactly the same except for when it comes
3.2. MIXTURE FILTERING USED IN ADAPTIVE PARTICLE KALMAN FILTER

To resampling, to the bootstrap filter. Since sampling from a continuous density elevates the problems of sample impoverishment, forming a continuous distribution from a discrete has to be accomplished. This is an old problem in statistics and is termed kernel density estimation (KDE). In KDE the problem is focused primarily around the choice of kernel bandwidth, in fact the actual choice of the kernel is not as important as is the bandwidth. Soon after its development, the optimal kernel and optimal bandwidth for the case of equal weights was derived. For different weight distributions this is still an open question. However forming kernel density estimators in high dimensions is not only a difficult problem but slow, thus almost immediately ruling it out for on-line use [5].

3.2 Mixture Filtering used in Adaptive Particle Kalman Filter

Mixture filtering concepts give the basis to use mixture densities to approximate posterior distributions, as well giving equations for mixture propagation, and maintenance. It should be stressed that these concepts assume a mixture with one type of density. In Adaptive Particle Kalman Filtering this assumption is relaxed, and several new questions need to be addressed. APKF can be seen as a mixture filtering problem where the mixture density is comprised of a Gaussian and discrete densities. Naturally it remains to answer how such a density could be propagated, how the mixture representation is computed and how it is maintained.

3.2.1 Resolving into Mixture of Gaussian Densities

Before any mixture filtering can be applied, the prior, process noise and observation noise densities need to be resolved into a mixture model, of a specific density.
3.2. MIXTURE FILTERING USED IN ADAPTIVE PARTICLE KALMAN FILTER

Since the concern here is with Gaussian mixtures, the algorithm utilized will be the classical approach to iteratively maximizing the log-likelihood function, the expectation maximization algorithm. The problem is formulated as uncovering the hidden class assignment for each data point, \( n \) here represents the label of each data point, \( N \) being the total number of data points, not to be confused with \( N_k \) which is the number of points during the algorithm assigned to the \( k^{th} \) cluster and they exist in \( j \) dimensional space, where \( k \), in contrast to the everywhere else thus far denotes the class assignment. The expectation step begins with computing the posterior for each point, with respect to each component density as initialized. From there, the maximization step seeks to refine estimates of the parameters, and for the special case of Gaussian mixture densities, the parameters can be written in a closed form relation to the posterior. Termination is signalled when the log-likelihood computed with the refined parameters falls below a set threshold for two subsequent set of parameters.

For reference, the algorithm is presented below, preceding which the parameters used in simulations are given in a table.

Table 3.1: Expectation Maximization simulation parameters

<table>
<thead>
<tr>
<th>Number of Clusters, ( K )</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Iteration</td>
<td>1000</td>
</tr>
<tr>
<td>Cov. Matrix Diagonal</td>
<td>False</td>
</tr>
<tr>
<td>Equal Cov. Matrices</td>
<td>False</td>
</tr>
<tr>
<td>Initialization for Mixing Proportions</td>
<td>Uniform, set to ( K^{-1} )</td>
</tr>
<tr>
<td>Initialization for Component Means</td>
<td>Draw randomly ( K ) data points</td>
</tr>
<tr>
<td>Initialization for Cov. Matrices</td>
<td>diag(( ... ), var(along ( j^{th} ) dimension), ( ... ))</td>
</tr>
</tbody>
</table>
3.2. MIXTURE FILTERING USED IN ADAPTIVE PARTICLE KALMAN FILTER

**Algorithm 5** Gaussian Mixture Estimation via Expectation Maximization [3]

1. Initialize values for the following quantities, $\mu_k$, $\Sigma_k$, and $\pi_k$

2. **E-Step**: Evaluate posterior with current parameters

   \[
   \gamma_{nk} = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{l=1}^{K} \pi_l N(x_n | \mu_l, \Sigma_l)}
   \]  

3. **M-Step**: Re-estimate parameters using current posterior

   \[
   N_k = \sum_{n=1}^{N} \gamma_{nk}
   \]

   \[
   \mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \cdot x_n
   \]

   \[
   \Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma_{nk} \cdot (x_n - \mu_k^{\text{new}})^T (x_n - \mu_k^{\text{new}})
   \]

   \[
   \pi_k^{\text{new}} = \frac{N_k}{N}
   \]

4. **Convergence**: Evaluate the log-likelihood and check for its convergence or in the parameters or terminate when maximum iterations reached.

   \[
   \ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln \left( \sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k) \right)
   \]

---

3.2.2 Mixture Propagation in APKF

Since the constituent components in APKF are the superposition of a linearized (either analytically or statistically) Kalman filter and a particle filter, propagation is
achieved through the parallel implementation of such filters. Given the initialization, where samples are drawn from the initial state distribution, $p(x_0)$, a choice exists as to when the time propagation should take place. The two options include whether clustering takes place before propagation or whether time propagation of the initial samples precedes clustering. In either scenario, measurement update will be proceed in a parallel fashion afterwards. As such the APKF will update the Gaussian component through extended Kalman measurement update, and the discrete samples via the particle filter update equations.

Another decision arises here, one that is related to clustering criterion. The moment preserving merge presented earlier, is used extensively in target-tracking where data association uncertainty introduces a exponential number of tracks during each scan, is certainly one option. Measurement ambiguity introduces several possible candidates as valid measurements and when some are disregarded, pruning of hypotheses is carried out, and can be thought of as collapsing components. Therefore however clustering is performed it must inevitably end with the moment preserving merge to form an estimate. In this context distance based reduction criteria are often favoured, with the clustering operation always proceeding from one mixture of Gaussian to another. In APKF many more degrees of adaptation can be conceived, and the conversions are enumerated below,

1. A single Gaussian component to L particles.

2. From L particles to a single Gaussian component.

3. Reducing an L component to an M component Gaussian mixture, where $M \leq L$, also known as merging.
4. Splitting an M component to an L component Gaussian mixture. Known as **splitting** and is an ill-posed problem.

5. Converting between L and M particles, which is known as **deterministic re-sampling**.

The final conversion, which will not be considered further was used by Li et al. [16] in an application for mobile robot localization, and performed quite well for this specific problem. The intuition was as follows, since the traditional resampling algorithms only consider particle weights, information contained in the spatial distribution of particles is ignored. Regardless of the fact that it seems very natural that particles within close proximity of each other must be representing the same state. A generalization called deterministic resampling was presented, where spatial distribution of particles is considered, and particles close to others are clustered into particles with a larger weight. Thus traditional resampling steps that follow, handle a smaller number of particles, fewer computational cycles are used. Another benefit is that no particles are discarded, while the particle diversity is maintained. A obvious downside is that the dimensionality free property of resampling no longer exists.

Items 1-4 of the enumeration constitute the scenarios that can be built into the APKF. Item 1 can be solved using sampling. Item 2, as the kernel density estimation problem. Item 3 and 4 are classically considered in Gaussian mixture filtering. Item 3 is crucial for the implementation of several tracking algorithms. While item 4 is a less explored option, since in GSF mixture components increase naturally, splitting is not a sought after operation. However it should be pointed out that in its generic form, Gaussian component splitting is an ill-posed problem, some type of qualification is required to result in a solvable form [26].
3.2. MIXTURE FILTERING USED IN ADAPTIVE PARTICLE
KALMAN FILTER

While enumerating possible adaptations, the control, which can be thought of as the signal that indicated when some kind of action which needs to be taken has not yet been specified. One reason for this is the difficulty of on-line dynamic model assessment. In [8] a nonlinear dynamic model assessment framework was presented using the Kolmogorov-Smirnov statistic. One possible method, is to constrain a certain quantity in the posterior approximation, for example restricting the number of Gaussian components used, thereby reducing a degree of freedom. Violation of this requirement then triggers the adaptation. An adaptation criteria would also be needed to understand the extent, and type of adaptation that is required.

Items 3 and 4 require an adaptation between two densities whose constituent parts are of the same type. An adaptation criteria is another name for a merging or splitting criteria, the goal of either is to find a similarity between components and apply an action accordingly. Merging methods which were reviewed first, splitting since the latter needs to be prefaced with more information will be skipped. In [6] several mixture reduction algorithms are surveyed and these were compared against a brute-force mixture reduction algorithm, which they derived. Their results showed that Runnalls Algorithm [24], which was also included as the first step of another algorithm, known as Gaussian mixture reduction via clustering (GMRC), performed the best in a practical tracking experiment [26].

Runnalls’ algorithm [24] works by a criteria that seeks to pick a pair of components, such that when these are merged the KL-divergence of the new mixture is minimized. The problem that arises is the intractability of the KL-divergence for Gaussian Mixtures. Hence Runnalls found an upper bound to the KL-divergence, which was used as the cost for merging two components labelled $i$ and $j$, where $w_i$
and \( w_j \) are the mixture weights and \( P_i, P_j, \) and \( P_{i,j} \) are the covariance matrices for the components under consideration along with the covariance matrix if they were to be merged,

\[
c_{i,j} = \frac{1}{2}((w_i + w_j) \log|P_{i,j}| - w_i \log|P_i|) - \frac{1}{2}w_j \log|P_j|.
\] (3.7)

The algorithm has a very simple structure, to initialize, set the current as the full mixture. Then calculate the \( c_{i,j} \) for all pairs in the mixture, the lowest cost pair is then merged using the moment preserving equations. Finally if the current mixture has the same number of components then the algorithm terminates. While this is the best algorithm (in terms of computations) if a single Gaussian is maintained its application is not necessary.

While the above discussion presented innovations that could have been included within APKF, due to time constraints on the author, and lack of familiarity with some theoretical results only a fraction of the available options were incorporated. Given time and a focused plan of action, it would not be inconceivable for all of the innovations to be deployed in an extension on the current work, and indeed several other paths are given in the final chapter. However for now the features utilized are described in detail alongside the rest of the filter, in the following section.

### 3.3 Building the Adaptive Particle Kalman Filter

Our algorithm was developed with the motivation of combining the benefits of the Kalman and particle filters. Naturally approaching a nonlinear filtering problem with Kalman assumptions requires a variant of either analytic or statistical linearization,
the most famous being the extended KF or the unscented KF for each method respectively. While both work well in practice, and are cornerstones of the community, in some cases the extended Kalman is known to diverge quite severely. Which filter is used however does not affect the implementation of our algorithm since, once the Gaussian component is identified, nonlinear Kalman and particle filtering can take place in parallel and hence whether the UKF or the EKF is used is rather immaterial from the algorithm’s point of view. It is hypothesized that the particles will overcome the difficulty, say an EKF experiences when handling a highly nonlinear model.

The major innovation in APKF is the assignment of particles in such a way so that a majority of them can be represented by a Gaussian, and a fraction of them remain unchanged. This process is termed component extraction and is described in detail. Component extraction is conceptually easy, from the EM algorithm a posterior value of the points conditioned on which class they belong to is given. Thus it is imperative to choose which cluster shall be termed Gaussian. This is achieved by finding the larger component weight, as this signifies greater contribution overall. Now the task of computing the posterior given either class is carried out, and at the end, two groups are labelled. Data points in the class belonging to those having a lower posterior probability conditioned on the component with a larger weight, are left untouched, henceforth these are known as particles. Those who are in the other category are deleted, since their posterior indicated a strong affinity for the Gaussian component, and are hence replaced by the component mean and covariance. Filtering is now carried out in parallel, and finally at the end of each cycle, the full estimate is found though a weighted combination of the two branches.

With the innovative feature of APKF expanded upon, the full algorithm cycle is
now presented. Notation is kept the same as the algorithms discussed while presenting those in the background. In the following chapter, simulations will be presented to show the relative performance of APKF. While estimation error is fundamental to consider, so too is a comprise that must be made, usually between error and additional computation. This comprise and several other considerations will be discussed after simulations are presented, and errors reported with a Monte Carlo regime of tests.
Algorithm 6 Adaptive Particle Kalman Filter

1: **Initialization:** Set the number of particles, $N$ and draw samples from the prior state density $p(x_0)$.

2: **Component Extraction:** Apply EM algorithm to samples setting the number of clusters to 2, and sort the output mixture weights in descending order. This results in $\pi = [\pi_1, \pi_2]$. Since $\pi_1$ is the largest mixture weight, this component mean and covariance matrix will be stored and used to initialize the EKF. If $\text{Prob (data point | Cluster 1)} > \text{Prob (data point | Cluster 2)}$ then the data point will be deleted, otherwise it will be stored as a particle.

3: **Time Propagation:** This step and all others following it are cycled through until the end of the simulation run. For Gaussian component initialize EKF using mean and covariance matrix belonging to component with weight, $\pi_1$. For particles use the particle filter time propagation equations.

4: **Measurement Update:** In the same fashion apply EKF measurement update to Gaussian component, and particle filter measurement update equations to particles. An additional step of resampling is required, solely for the particles.

5: **Combining Filter Outputs:** As a final step the outputs from the two filters are combined resulting in an approximation to the filtering density mean as follows, noting $N_p$ is the effective number of particles used:

$$\mathbb{E}[x_k | Z_k] \approx \pi_1 \cdot \hat{x}_k^+ + \frac{\pi_2}{N_p} \cdot \sum_{i=1}^{N_p} x_{k,i}^+.$$  \hspace{1cm} (3.8)
Chapter 4

Results and Evaluation: Assessing the Performance of APKF

4.1 Introduction

In this chapter the proposed algorithm will be used on two classical benchmarks in the nonlinear filtering literature. The fundamental goal of the APKF is to reduce the number of particles used, but not their effect. Thus with a reduced number of particles, the particle filter can be applied to energy and computationally sensitive applications. Clearly performance cannot suffer too greatly due to reduction in number of particles, therefore the root mean square error will be used as the indicator of estimator quality. Computational time will not be explicitly noted, neither as computational complexity nor overall running time. This is because our goal to present the case that a significant reduction in particles can be achieved. However it is quite natural that the component extraction step will slow down the overall runtime of the algorithm. This need not be a issue because with fewer particles, successive filtering steps should be executed much faster. The number of particles, as well as the number of reduced particles will be reported.
4.2 Application Examples: Overview

In recursive Bayesian estimation several problems serve as benchmarks and are familiar to practitioners. One might look negatively upon these traditional applications as being trivial or outdated, but they occur so often in the literature that not including them would be a questionable move, at best. Also owing to the extremely wide applicability of stochastic filtering, when fundamental new algorithms are developed, for the ease of non-specialists these examples are a litmus test of their quality.

Two examples are widely used in the literature, the first being the single state, univariate non-stationary growth model, which is defined by the following state and observation equations.

**Definition 4. Univariate Non-stationary Growth Model**

\[
x_k = \frac{1}{2} x_{k-1} + 25 \frac{x_{k-1}}{1 + (x_{k-1})^2} + 8 \cos(1.2(k - 1)) + w_k
\]

\[
z_k = \frac{(x_k)^2}{20} + v_k
\]

\[
w_k \sim \mathcal{N}(0, 1) \quad (4.3)
\]

\[
v_k \sim \mathcal{N}(0, 1) \quad (4.4)
\]

The next example, which is a classical problem in target tracking, is the bearings-only tracking benchmark. Two sensors measure the bearing angle with respect to their position, and a single target is moving on an unknown trajectory. While not addressed herein, more complicated multiple target problems can be decoupled into single track tracking problems, hence this example is widely studied.

**Definition 5. Bearings-only Tracking with 2 Sensors** The state vector is the
4.2. APPLICATION EXAMPLES: OVERVIEW

following,

\[ x = [x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^T. \] (4.5)

With the following dynamical equations,

\[
\begin{bmatrix}
  x_k \\
  y_k \\
  \dot{x}_k \\
  \dot{y}_k
\end{bmatrix}
= \begin{bmatrix}
  1 & 0 & \Delta t & 0 \\
  0 & 1 & 0 & \Delta t \\
  0 & 0 & 1 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
  x_{k-1} \\
  y_{k-1} \\
  \dot{x}_{k-1} \\
  \dot{y}_{k-1}
\end{bmatrix}
+ \mathbf{w}_k \] (4.6)

where the \( \mathbf{w}_k \) is zero mean Gaussian noise process and the sampling period be 0.01.

The measurement model is described as follows, where \( i \in \{1, 2\} \):

\[ z^i_k = \arctan\left(\frac{y_k - s^i_y}{x_k - s^i_x}\right) + v_k \] (4.7)

where \( v_k \sim \mathcal{N}(0, \sigma^2) \).

Table 4.1: Simulation parameters for benchmarks

<table>
<thead>
<tr>
<th></th>
<th>UNGM</th>
<th>Bearings-only Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Density Mean Vector</td>
<td>0.1</td>
<td>[0, 0, 0, 0]</td>
</tr>
<tr>
<td>Initial Density Cov. Matrix</td>
<td>1</td>
<td>diag(0.1, 0.1, 10, 10)</td>
</tr>
<tr>
<td>Number of Particles. N</td>
<td>1, 000</td>
<td>25, 000</td>
</tr>
<tr>
<td>State Dimension</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Length of Simulation, samples</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>Measurement Noise, ( \sigma )</td>
<td>1</td>
<td>0.05 rad.</td>
</tr>
<tr>
<td>Sampling Frequency</td>
<td>N/A</td>
<td>0.01</td>
</tr>
<tr>
<td>Sensor Location</td>
<td>N/A</td>
<td>( s^1 = [-1, -2]; s^2 = [1, 1] )</td>
</tr>
</tbody>
</table>
This application is a very difficult estimation problem. Part of this stems from
observability issues, which leads to diverging estimates, to safeguard against this, two
sensors are used instead of one. The trajectory will include turns and hence this
is a manoeuvring tracking problem. Since the goal of this thesis is to reduce the
number of particles used, tracking performance should be diminished, but in fact it
will be shown that the diversity of using a Gaussian nonlinear component alongside
particles in some cases prevents the position estimate from diverging during unstable
linearization of the measurement model.

When a similar problem was used in Gordon’s seminal work [10], a technique called
roughening was utilized to reduced the number of particles. This was needed due to
issues within the bearings-only setup, when implemented with the regular bootstrap
filter it tends to diverge rather frequently. Thus the added benefit of the APKF is
that estimation error does not tend to grow, a hallmark of a diverging particle filter.
To further emphasize this, only the bootstrap filter will be used in the simulations,
and as part of the APKF, so that the comparison is sound.

4.3 Experimental Regime: Tests, Quantities, and Data Reported

4.3.1 Gaussian Sum Based Estimation:

Now that two benchmark examples have been introduced, experimentation will
commence with the estimation techniques based on Gaussian sum filtering methods.
Recall that GSF techniques are important solutions to cases where the process and
observation noise is non-Gaussian, but still additive. GSF methods are clearly non
particle based, and thus the analysis for additive Gaussian nonlinear systems is equiv-
alent to the choice of local filter utilized. In this applications the local filter used was
the EKF, since this is also used in the APKF.

4.3.2 Particle Based Estimation:

Particle filters are often seen as the holy grail of nonlinear state estimation, however features such sample impoverishment and delay time make their implementation nearly impossible in several situations. For example, if particle diversity is not promoted by some means, such as resampling, the filter degenerates. It seems counter intuitive though claiming that on the one hand resampling alleviates degeneracy and then saying that it can cause it as well. What resampling does in actuality is reduce the diversity in particles, thus many particle end up with the same value. Resampling therefore does not promote diversity. Regularization is seen as one prevalent solution, noting its own high complexity.

One naive attempt to circumvent this phenomenon would be to choose a sufficiently large number of particles. This backfires because as state dimension increases, the number of particles needed scales exponentially, thus the problem of impoverishment to a certain degree would still persist. This is the reason why regularization is seen as such an effective technique, by utilizing a kernal density estimate, a continuous distribution is resampled from, instead of the traditional which encourages impoverishment.

4.4 Simulation Plots and Results:

4.4.1 Univariate Non-stationary Growth Model

Three plots have been selected to showcase results. The first plot is the state estimate compared to the actual state, for the three filters. Second, the RMSE over
100 Monte Carlo runs is plotted, indicating that a significant reduction in particles still leads to a viable estimate. Finally a plot displaying the number of particles used in the APKF is presented, showing that on average 386 particles combined with an EKF lead to a better estimate than the single EKF.

Figure 4.1: Sample Trajectory with Filter Estimates
4.4. SIMULATION PLOTS AND RESULTS:

Figure 4.2: RMSE Over 100 MC runs, $N = 1,000$, $N_{APKF,avg} = 386$

Figure 4.3: Percentage of Total Particles used in APKF
4.4. SIMULATION PLOTS AND RESULTS:

4.4.2 Bearings-only Tracking with 2 sensors

Three plots have been presented, in decreasing probative value. Firstly, the position RMSE error, over 100 Monte Carlo trials is shown averaged over each time step, this is important because a single RMSE value is not recorded for an entire trajectory, which can lead to misinterpreting the overall estimation error over different portions of the trajectory. Second, a degenerate simulation run is shown, and then within the same run, a turn is zoomed into.

![Figure 4.4: RMSE Over 100 MC runs per timestep, $N = 25,000$, $N_{APKF,avg} = 4,997$](image)

Figure 4.4: RMSE Over 100 MC runs per timestep, $N = 25,000$, $N_{APKF,avg} = 4,997$
4.4. SIMULATION PLOTS AND RESULTS:

Figure 4.5: Sample Trajectory with Filter Estimates

Figure 4.6: Trajectory Turned, zoomed, with Filter Estimates


4.5. ANALYSIS & DISCUSSION:

When discussing estimator quality, it needs to be predicated on the understanding that the dimension of the state-space increases computational complexity. Higher dimensional state-spaces, such as in the bearings-only tracking example naturally burden estimation algorithms more, this is also evident by noting that in the case of particle filters, particles reside in a four dimensional space rather than in one dimension, as with the UNGM. Therefore it is not valid to boast lower estimator error, without stating this fact.

A secondary feature in any estimation problem is to point out the linearity of the state and observation equations. While this was done earlier, it is illustrative to go over this again. Nonlinearity in both equations results in a system which is much harder to estimate, because twice the non-linearity is introduced when compared to an state-space like the bearings-only tracking, where only the observation is nonlinear. While it is not important delve in to degrees of nonlinearity, comparing a squared term with an arctan(·) term, both distort observables considerably. Pairing this with state dimension these two features of a problem indict the difficulty in accurate state estimation.

4.5.1 Reflection on Univariate Nonlinear Growth Model Estimation with APKF:

UNGM is a 1-state estimation problem, hence if any algorithm would not beat, say the EKF estimate for such a problem, it would not be worthy of study. The APKF passes this test, and shows that a parallel combination of an EKF with a reduced number of particles bootstrap filter does much better, in terms of RMSE than the
4.5. ANALYSIS & DISCUSSION:

EKF. It also achieves proximity to the particle filter error, when the full number of particles is used, which it is important to note uses 60% more particles. The system is also of great importance in nonlinear time series analysis, so the results here have wide implications, and can be applied to several different fields.

4.5.2 Reflections on Bearings-Only Tracking with APKF:

Bearings-only tracking is an inherently engineering application, a simplified sensor model along with linear state dynamics are used to investigate the performance of filtering algorithms. The crux of the APKF is the initial EM-clustering operation, which is used to resolve an early posterior density into a mixture of Gaussian density. For this application the posterior lives in 4-D state space and hence already signals trouble for an iterative nonlinear optimization procedure such as EM. Hence when the performance of the APKF is viewed from this point of view, in a multi-sensor and manoeuvering target example, the results are quite significant.

From the viewpoint of particles used, the APKF uses on an average only a fifth of the original particle cloud, which itself resides in a 4-D state-space, and can yet offer fairly accurate tracking. While the EKF does well in this problem, it is always prone to divergence, at least more easily than the particle filter, thus using the APKF offers this benefit as well.
Chapter 5

Directions for Further Inquiry & Conclusion

5.1 Future Work

5.1.1 Realistic Sensor Models

All the sensor models used in this thesis were considered to be ideal. Quantitatively, this is described by the situation where the probability of detection ($P_D$) is unity and the probability of false alarm ($P_{FA}$) is nil. Realistic sensor models can generally be classified as sensors whose $P_D < 1$ and $P_{FA} > 0$.

If realistic surveillance environments are used in the testing, clutter plays a huge role. Connected to the false-alarm rate, clutter is essentially those false-alarms that seem like the target. Overcoming clutter requires the resolution of what is called, measurement origin uncertainty, and is done through data association [11].
5.1. FUTURE WORK

5.1.2 High Dimensional State-Space

In practical navigation and positioning applications it is not unheard-of to deal with estimation problems where 9 to even 15 states are utilized. Such high dimensional problems presents issues computationally different from lower dimensional systems. Algorithms that work in lower dimension are seldom tractable when dimensionality is increased, and so a methodology is required to deal with such problems [18].

5.1.3 Multi-Sensor Multi-Target Extension

The prevalent paradigm in target tracking, which constitutes a large part of the nonlinear filtering community, is to view their central problem as a multi-sensor multi-target scenario. As such algorithms for state estimation must be able to deal with multiple sensor data, and the spawning or vanishing of one or several targets. Extending a state-estimation algorithm to accommodate these relaxations is not trivial [4].

When multiple sensor data is received, apart from practical considerations that need to be thought of when working with realistic sensors, thought has to be given to how data from these multiple sensors will be combined. Two major approaches are the Bayesian, or classical probabilistic fusion, and the Dempster-Shafer or evidential reasoning fusion [7].

Presence of multiple and vanishing targets truly presents a difficult issue. If the number of targets is changing, so too will the state dimension and Bayesian statistics cannot handle such a situation. Finite set statistics will have to be used instead, and is the accepted methodology to approach this problem [22].
5.2 Conclusion

Guiding this thesis was the pursuit to answer one question, how can the computational burden of the particle filter be lessened, so as to encourage its use in real-time applications. Our approach was based on using information contained in the spatial distribution of particles to reveal some underlying structure which could be exploited. Said underlying structure could perhaps lead to an efficient way of processing the filtering density, where an abundance of particles might be compensating the unavailability of a parametric filter. Before such an approach could be implemented all the necessary background needed to be searched and condensed, to understand were the literature was in terms of developing similar ideas. This task was accomplished in Chapter 2, where the two underlying paradigms for nonlinear filtering were reviewed to achieve a solid foundation before new ideas could be developed on top. This chapter also served to generalize two of the most commonly used algorithms in nonlinear filtering, the extended Kalman filter and the particle filter which were presented in the introduction.

Throughout Chapter 2, hints in the literature were sought which pointed to this line of innovation. The Gaussian sum filter being the oldest innovation that forced a structure onto the filtering density and thereby achieving greater exploratory ability in the state space with modest complexity [1]. Unfortunately this filter was the extent for this line of inquiry, bogged down by computing power available in the early 1970s, it was not until the particle filter that a remarkably new approach to nonlinear filtering was offered. Particle filtering shared a similarly strange history, its basic approach was known to physicists for decades but this time not only lacking computing power but also a degeneracy issue hindered widespread utilization. Both algorithms worked
with a structure imposed on underlying densities, updating this structure when measurements were validated. Of course neither algorithm was without additional issues, such as what to do with extra pieces of structure that during the run cost computation but offered very little probative value. This and several other questions were outlined, in addition to separating which questions our approach would have to respond to, in Chapter 3.

In Chapter 3 the proposed filtering algorithm was developed, through answering the requirements outlined as necessary. In short the basic idea of these requirements were taken from the area of mixture filtering, and include mixture resolution, propagation, and maintenance [30]. Having considered other possible innovations, a subset of those described were implemented, and an algorithm presented whose goal lined up with the thesis statement. Having presented an algorithm it was now time to test it against benchmarks, and all of Chapter 4 was devoted to it, as well as offering some analysis as to its behavior. The simulations showcased, acknowledged the added burden required in pre-filtering mixture representation, but did not dwell explicitly on time complexity. Instead aiming to show heuristically that a reduction in the number of particles used would positively affect the time required in all steps thereafter. The estimator performance was judged in terms of RMSE, and was a compromise between the particle filter and EKF for the 1-dimensional benchmark. However the resulting RMSE for the 4-dimensional benchmark clearly showed room for improvement even though a significant reduction in particles was achieved. Such performance was accredited to the specific mixture representation technique experiencing difficulty in high dimensional space. Some other explanations related to overall nonlinearity of a system were touched upon, but nothing conclusive can be said about this.
The adaptive particle Kalman filter (APKF) utilizes the spatial distribution of the initial particle cloud to achieve a unique type of filter. One which significantly reduced the effective number of particles needed, and worked exceptionally well for the 1-dimensional benchmark. Finding a Gaussian and particle component structure was the key, and accurately allotting portions of the filtering density likewise. The goals of the thesis statement are all achieved within this one filter, it is hoped that further development on this novel idea can improve its function even more. Several directions where extensions can be made were outlined in this concluding chapter.
Bibliography


[19] Bin Liu, Xiao chuan Ma, Cheng peng Hao, Chao huan Hou, and Mei Li. A bearings-only-tracking framework based on the ekf and ukf combined algorithm.


